Additional slides

Summary (of Combinatorial Optimization)

- MI and MR are combinatorial optimization problems
- Simple cost model
- Best first search
- Conflicts dramatically focus search
- Decrease agenda size

Summary (of LTMS)

- An LTMS incrementally maintains the consequences of a propositional theory
- Using the LTMS
  - MI : to check if a mode assignment is consistent with the observations
  - MR : to check if a mode assignment entails the goal
- Propositional inference using unit propagation
  - produces well-founded support
- Dependency structure used to
  - identify a conflict from an inconsistent theory
  - incrementally delete clauses

Summary (of Dynamics)

- Explicit model of mode transitions
  - models commanded transitions, failure transitions, repair transitions, intermittency
  - probability and cost of transitions
- Concurrent transition system models
  - represented using a propositional temporal logic
  - modeling language simplifies specification
- MI and MR reformulated using transition systems

Model-based programming language

```
(defcomponent valve (?name)
  :attributes ((sign-values (flow (input ?name)))
               (sign-values (flow (output ?name)))
                ...)
  ...
  (closed
   :model (and (= (flow (input ?name)) zero)
               (= (flow (output ?name)) zero))
   :transitions ((open-valve
                  :when (open (cmd-in ?name))
                  :next open)
                 (:otherwise :persist)))
  ...)
```

Features of programming language

- Object-oriented
  - component-based and compositional
- Strong typing
- Qualitative modeling
  - sign algebra, relative values, …
- Explicit closure on transitions for frame axioms
- Automatically compiles into a propositional temporal logic specification